



Graduado en Ingeniería Informática

Universidad Politécnica de Madrid

Escuela Técnica Superior de Ingenieros Informáticos

TRABAJO FIN DE GRADO

Desarrollo de un gestor de plantillas de mensajes
HL7 para la carga de datos clínicos

Autor: Israel Suárez Santiago

Tutor: David Pérez del Rey

MADRID, JUNIO DE 2014

Índice de contenidos

i. RESUMEN DEL TRABAJO	iv
i.i Resumen	iv
i.ii Abstract.....	iv
1. INTRODUCCIÓN	1
1.1 Planteamiento del problema.....	1
1.2 Objetivos	1
1.3 Solución propuesta.....	1
2. ESTADO DE LA CUESTIÓN	3
2.1 Interoperabilidad en sistemas biomédicos	3
2.2 Trabajos previos.....	3
3. MIRTH CONNECT	4
4. ESTÁNDARES Y VOCABULARIOS	5
4.1 Health Level Seven (HL7)	5
4.2 Clinical Document Architecture (CDA)	5
4.3 Vocabularios clínicos.....	6
5. SERVICIOS Y HERRAMIENTAS	6
5.1 Base de datos de conceptos clínicos	7
5.2 Autocomplete Service.....	7
5.3 Xmlint.....	8
5.4 Graphviz.....	9
5.5 Librerías de Javascript	10

5.5.1 JQuery	10
5.5.2 Fancybox	11
5.5.3 Bootstrap	11
5.5.4 Codemirror	11
6. LENGUAJES DE PROGRAMACIÓN	12
6.1 Nodejs	13
6.2 Otros lenguajes	13
7. ANÁLISIS DEL SISTEMA	14
7.1 Requisitos.....	14
7.2 Casos de uso.....	19
8. DISEÑO E IMPLEMENTACIÓN.....	22
8.1 Funcionalidad.....	22
8.2 Estructura interna	23
8.3 Interfaz de usuario	25
9. PRUEBAS Y RESULTADOS	29
10. CONCLUSIONES	31
BIBLIOGRAFÍA	32
ANEXOS	34
Anexo 1 – Ejemplo básico de plantilla	34
Anexo 2 – Ejemplo de mensaje HL7v3 con datos.....	35



i. RESUMEN DEL TRABAJO

i.i Resumen

El trabajo presentado a lo largo de este documento es el resultado del TFG¹ realizado por Israel Suárez Santiago, alumno de la Escuela Técnica Superior de Ingenieros Informáticos (ETSIINF) de la Universidad Politécnica de Madrid (UPM). Dicho trabajo tiene como finalidad proporcionar una herramienta que, basada en estándares previamente estudiados, permita la fácil creación y gestión de plantillas de mensajes HL7v3² a las que posteriormente se le añadirán datos clínicos que serán insertados en una base de datos para su fácil acceso y consulta.

La herramienta desarrollada únicamente facilita una serie de opciones para la creación de la plantilla en sí, que servirá como base para la creación de mensajes HL7v3, es decir, no permite la inclusión de datos específicos en las plantillas generadas, que deberá hacerse con alguna herramienta externa o bien manualmente.

Las plantillas generadas por la herramienta se basan principalmente en el estándar CDA³, que proporciona una amplia guía para la correcta generación de mensajes HL7v3. La herramienta garantiza que las plantillas resultantes estarán correctamente formadas, siendo acordes al estándar anteriormente citado y siendo, además, sintácticamente correctas, es decir, el documento *.xml* generado no contendrá errores.

i.ii Abstract

This document is the result of the TFG developed by Israel Suárez Santiago, student of *Escuela Técnica Superior de Ingenieros Informáticos (ETSIINF)* of the *Universidad Politécnica de Madrid (UPM)*. This work aims to offer a tool based on standards that can facilitate and manage the creation of HL7v3 templates. Clinical data will be added to those templates in order to load them into a database and query them fast and easily.

¹ TFG: Trabajo de Fin de Grado

² HL7v3 (Health Level Seven Version 3): Este estándar se explica en la sección 4.1 *Health Level Seven (HL7)*.

³ CDA (Clinical Document Architecture): Este estándar se explica en la sección 4.2 *Clinical Document Architecture (CDA)*.



The tool only facilitates several options to create the template, that will be used to generate the HL7v3 messages, but it does not permit the inclusion of data on them. The inclusion of data will be done manually or using an external tool.

The generated templates are based mainly on the CDA¹ standard, that provides a widely guide to create HL7v3² messages. The tool guarantees that the resulting templates have been correctly generated, following the previous standard and with no errors in the *.xml* document generated.



1. INTRODUCCIÓN

1.1 Planteamiento del problema

El trabajo que aquí se presenta surge tras la necesidad de compartir datos clínicos dentro del proyecto europeo EURECA⁴. Una de las partes más importantes de este proyecto consiste en la capacidad de compartir datos clínicos basándose en estándares, de ahí que se decida compartir dichos datos usando mensajes HL7v3.

Sin embargo, el uso de este tipo de mensajes trae consigo la dificultad de generar correctamente dichos mensajes, ya que es necesario tener un amplio conocimiento acerca de estándares como CDA o HL7v3 para su correcta creación, pues es necesario tener en cuenta no sólo la sintaxis de los mismos, sino que también es imprescindible conocer cómo expresar adecuadamente la información clínica que se posee dentro del mensaje. Estos dos factores se hacen imprescindibles a la hora de cargar la información contenida dentro de los mensajes en una base de datos, ya que dicho proceso se hace automáticamente y es necesario tanto que los mensajes no contengan errores, como que la información se encuentre donde se espera que esté.

1.2 Objetivos

El objetivo que se desea conseguir con este trabajo es facilitar de algún modo la generación de mensajes HL7v3, tanto si se tiene un gran conocimiento de dichos mensajes como si se desconoce por completo cómo generarlos. Facilitando la generación de dichos mensajes se facilita y acelera el intercambio de datos clínicos. Además, el uso de este tipo de mensajes para compartir datos hace que dicho proceso sea fácilmente escalable, ya que los mensajes que están basados en estándares ampliamente aceptados.

1.3 Solución propuesta

La solución propuesta consiste en el desarrollo de una herramienta web que permita crear plantillas para mensajes HL7v3 de forma sencilla, de manera que sólo

⁴ EURECA: Enabling information re-Use by linking clinical REsearch and Care



habría que añadir los datos que se deseen dentro de dichas plantillas. El proceso de rellenar las plantillas con datos puede automatizarse con herramientas externas y *open source*, como es el caso de *Mirth Connect* (ver sección 3. *Mirth Connect*), que, dada una base de datos con datos referentes a pacientes y una plantilla (generada con la herramienta desarrollada), es capaz de colocar en dichas plantillas los datos procedentes de la base de datos en los campos del mensaje que se deseen, facilitando y acelerando en gran medida el proceso de intercambio de datos clínicos usando mensajes HL7v3.

Las plantillas generadas se irán componiendo a base de añadir bloques atómicos (fragmentos de código XML) a un esquema de mensaje básico. La herramienta facilitará al usuario la creación de las plantillas aportándole información acerca de los bloques que puede usar con un concepto clínico dado o bien informándole de los bloques que puede insertar dentro de una sección concreta de la plantilla. De esta manera se asegura que la plantilla resultante no sólo es correcta sintácticamente, es decir, el documento *.xml* no contiene errores, sino que además, dicha plantilla estará correctamente formada siguiendo los estándares ya citados.

La herramienta propuesta podría ser de gran utilidad en cualquier proyecto o trabajo en el que fuera necesario el uso de mensajes HL7v3 para intercambiar datos, sin embargo, dicha herramienta será utilizada en primera instancia dentro del proyecto EURECA⁴, que es un proyecto europeo llevado a cabo por varias entidades, entre la que se encuentra la ETSIINF de la UPM y cuyo objetivo principal es proporcionar una plataforma segura y escalable para el intercambio de datos clínicos procedentes de EHR's⁵, además de establecer una interoperabilidad semántica entre dichos EHR's y *clinical trials* (ensayos clínicos).

Se puede encontrar más información acerca de la herramienta desarrollada en las secciones 7.1 *Requisitos*, 8.1 *Funcionalidad* y 8.3 *Interfaz de usuario*.

⁵ EHR: Electronic Health Record



2. ESTADO DE LA CUESTIÓN

2.1 Interoperabilidad en sistemas biomédicos

La informática biomédica es una ciencia que aúna conocimientos de informática y medicina y cuyo objetivo es la digitalización, almacenamiento, intercambio y recuperación de datos clínicos para la solución de problemas y la toma de decisiones de una forma más rápida y eficiente.

Las TIC⁶ juegan un papel muy importante dentro la biomedicina, teniendo una gran cantidad de aplicaciones dentro de la misma, como la elaboración de estándares y codificación de datos clínicos, el desarrollo de bases de datos o el almacenamiento del historial clínico de pacientes. El uso de las nuevas tecnologías dentro de la medicina también ha facilitado la interoperabilidad entre sistemas biomédicos, haciéndose muy importante el intercambio de datos e información, no sólo sobre pacientes, sino también sobre enfermedades o medicamentos. Este hecho supone una gran ventaja tanto para el personal médico como para los pacientes, ya que, al tener a su alcance mucha más información, es posible conocer en más profundidad algunos aspectos, como los efectos que producen algunos medicamentos o tratamientos en pacientes con determinadas características, información sobre ensayos clínicos o tratamientos experimentales o el acceso al historial completo de un paciente

Esta necesidad de almacenar, compartir y consultar este tipo de datos de una manera eficiente hace que vayan surgiendo nuevos estándares que sirven como guías para realizar de una forma más óptima dichas acciones. Algunos de estos estándares están incluidos en HL7 y pretenden unificar la forma en la que se intercambian datos clínicos.

2.2 Trabajos previos

Actualmente existen algunos programas, servicios o guías que facilitan de alguna manera la creación y manejo de mensajes HL7v3, sin embargo, no existen herramientas que permitan la creación de mensajes (o plantillas) utilizando bloques atómicos, como hace la herramienta desarrollada en este TFG.

⁶ TIC: Tecnologías de la Información y Comunicación



Una de las herramientas que puede ser utilizada como un gestor de plantillas es ART-DECOR⁷. Esta herramienta cuenta tanto con una aplicación de escritorio, desarrollada en Java, como con una interfaz web que es accesible de forma gratuita. En la versión web se pueden gestionar una serie de plantillas basadas en HL7 y CDA, así como algunos ejemplos de código XML con datos de ejemplo.

Además, en la [web oficial de HL7](#) pueden encontrarse una serie de herramientas para la gestión de mensajes basados en CDA, así como gran cantidad de documentación.

De esta forma, se podría decir que, aunque hay varios proyectos o herramientas previas en los que se hace uso de plantillas para la generación de mensajes, no existe (o no he conseguido encontrar) ninguna herramienta que permita la creación de plantillas en la misma forma que lo hace la herramienta desarrollada.

3. MIRTH CONNECT

Mirth Connect, como ya se ha mencionado, es un software *open source* que podría utilizarse como complemento a la herramienta desarrollada en este TFG para generar los mensajes HL7v3 en sí. Este software está principalmente desarrollado para trabajar con mensajes HL7 (versiones HL7v2 y HL7v3) y puede considerarse como una herramienta bastante útil para insertar datos en las plantillas que genera la herramienta descrita en este documento.

El programa se comporta como una tubería, donde en un extremo se introducen datos en un formato y en el otro se obtienen datos en el formato deseado. Mirth Connect es capaz de interpretar datos de diversas fuentes, como bases de datos, documentos XML, CSV..., por lo que puede ser utilizado en gran cantidad de entornos junto con las plantillas generadas por la herramienta para obtener mensajes HL7v3 basados en dichas plantillas y con los datos cargados en las mismas, “independientemente” de la forma en la que los datos estén almacenados en la fuente de datos.

La combinación de Mirth Connect con la herramienta desarrollada puede acelerar en gran medida el proceso de intercambio de datos entre distintas entidades,

⁷ ART-DECOR: Advanced Requirements Tooling - Data Elements, Codes, OIDs and Rules



además de facilitar dicho proceso, ya que abstrae en gran medida al usuario de la complejidad de generar los mensajes y almacenar los datos que se poseen en ellos.

4. ESTÁNDARES Y VOCABULARIOS

4.1 Health Level Seven (HL7)

HL7 International es una organización sin ánimo de lucro fundada en 1987 y que se encarga de desarrollar estándares para el intercambio, integración y recuperación de información clínica almacenada en formato digital. Su objetivo es crear el mejor y más ampliamente usado conjunto de estándares en el campo de la medicina. Dentro del gran conjunto de estándares desarrollados por HL7 se encuentran HL7v3 y CDA® Release 2, utilizados tanto para crear los bloques atómicos de texto XML como las plantillas de mensajes HL7v3 en sí.

4.2 Clinical Document Architecture (CDA)

Como se ha indicado en el apartado anterior, CDA es uno de los estándares desarrollados por HL7 International, que especifica la estructura y sintaxis de los *clinical documents* (en este caso, mensajes HL7v3) utilizados para el intercambio de datos clínicos. Según el estándar, los documentos generados siguiendo dicho estándar tienen las siguientes seis características:

1. Persistencia
2. Administración
3. Potencial de autenticación
4. Contexto
5. Totalidad
6. Legibilidad (humana)

Además, este estándar utiliza un lenguaje de marcado, es decir, utiliza etiquetas para delimitar el inicio y fin de un elemento dentro del texto. El lenguaje utilizado es



XML⁸, pero incorporando sus propias etiquetas y sus propios tipos complejos de datos. Un ejemplo de mensaje HL7v3 basado en CDA puede verse en *Anexo 2 – Ejemplo de mensaje HL7v3 con datos*.

4.3 Vocabularios clínicos

Una de las opciones proporcionadas por la herramienta consiste en filtrar los bloques atómicos que pueden ser utilizados con un concepto clínico dado. Para implementar esta opción se ha utilizado una base de datos con conceptos clínicos previamente creada por trabajadores del Grupo de Informática Biomédica (GIB) de la ETSIINF y utilizada por otros servicios.

En esta base de datos se encuentran conceptos pertenecientes a tres vocabularios clínicos distintos que cubren gran parte de los conceptos clínicos que pueden ser utilizados y que van siendo actualizados con nuevos conceptos. Los tres vocabularios son:

- SNOMED: Systematized Nomenclature Of Medicine Clinical Terms
- LOINC: Logical Observation Identifiers Names and Codes
- HGNC: HUGO Gene Nomenclature Committee

5. SERVICIOS Y HERRAMIENTAS

En esta sección se indican una serie de herramientas o servicios que utiliza la herramienta desarrollada para llevar a cabo algunas de sus opciones. Es necesario indicar que las siguientes herramientas no han sido implementadas por el alumno, simplemente han sido instaladas o utilizada alguna de sus funcionalidades para añadirlas a la herramienta desarrollada.

⁸ XML: eXtensible Markup Language



5.1 Base de datos de conceptos clínicos

Como se ha mencionado en la sección anterior, ha sido necesario el uso de una base de datos con información sobre conceptos clínicos para implementar algunas de las opciones de la herramienta. Esta base de datos contiene, además de los conceptos, información sobre los mismos, que es utilizada dentro de la herramienta para indicar al usuario los bloques que puede utilizar con un concepto dado. Cada uno de estos conceptos tiene las siguientes características:

- Puede pertenecer a una o varias de las siguientes clases: “*Entity*”, “*Procedure*”, “*Observation*” o “*SubstanceAdministration*”
- Dentro de estas clases puede tener los siguientes atributos: “*Code*”, “*MethodCode*” y “*TargetSiteCode*”

Atendiendo a las características anteriores se determina los bloques que pueden ser utilizados con un concepto concreto. Para los conceptos que pertenecen a la clase “*Observation*” o “*SubstanceAdministration*”, la base de datos contiene información acerca de si dichos conceptos pueden tomar valor o no. Esta información también se ha utilizado para poder filtrar los bloques a utilizar con un concepto.

A continuación se muestra una tabla con información relevante acerca de los datos incluidos en la base de datos anterior:

Vocabulario	Nº Conceptos	Entities	Procedures	SBADM ⁹	Observations
SNOMED	294.953	80.936	81.192	2.029	195.626
LOINC	61.098	0	0	0	61.098
HGNC	33.605	0	33.605	0	33.605
Total:	389.656	80.396	114.797	2.029	290.329

Tabla 1 – Clases de conceptos

5.2 Autocomplete Service

*AutocompleteService*¹⁰ es un servicio web que, aunque no ha sido implementado por el alumno, sí que ha sido mantenido y actualizado por el mismo. Este servicio web es utilizado dentro de los proyectos europeos EURECA e INTEGRATE¹¹.

⁹ SBADM: SubstanceAdministration



El servicio web se utiliza principalmente para obtener sugerencias de conceptos atendiendo a las clases de los mismos, aunque también es utilizado para anotar los conceptos que pueden tomar valor mediante el escaneo de bases de datos con datos clínicos y es ésta última funcionalidad la que se ha usado dentro de esta herramienta.

El servicio utiliza la base de datos indicada en el apartado 4 de este mismo documento, y es en esta misma base de datos donde anota la información referente a los conceptos que pueden tomar algún valor, el rango de valores que puede tomar y las unidades de dichos valores. Dentro de la herramienta se utiliza esta información para saber si se pueden utilizar los bloques de “*Value*” (si el concepto pertenece a la clase “*Observation*”) o “*DoseQuantity*” (si el concepto pertenece a la clase “*SubstanceAdministration*”).

El WSDL¹² del servicio web se encuentra en la dirección <https://kandel.dia.fi.upm.es:8443/AutocompleteService/services/AutocompleteService?wsdl>, aunque para utilizarlo es necesario tener un certificado.

5.3 Xmlint

Xmlint es un programa para línea de comandos en Linux que está contenido en el paquete “*libxml2*” y que sirve para parsear texto en formato XML. Este programa ha sido utilizado dentro de la herramienta para *indentar*¹³ el código XML de la plantilla antes de mostrarlo al usuario. Cuando el usuario utiliza la opción de visualizar el texto, se crea en el servidor un fichero XML con el contenido de la plantilla completa o el bloque que ha seleccionado el usuario para, posteriormente, usar el contenido de dicho fichero como entrada a xmlint. La salida de dicho comando será lo que se muestre por pantalla, que junto con el uso de distintos colores en el texto, facilita en gran medida la interpretación y legibilidad del código XML al usuario.

¹⁰ AutocompleteService: Servicio web para el autocompletado o filtrado de conceptos clínicos y la anotación de conceptos que pueden tomar algún valor.

¹¹ INTEGRATE: <http://www.fp7-integrate.eu/>

¹² WSDL: Web Services Description Language

¹³ Indentación: anglicismo utilizado en informática que hace referencia a la tabulación del código atendiendo al bloque de texto al que pertenece, facilitando de esta manera la legibilidad del código.



5.4 Graphviz

Graphviz es un software *open source* utilizado para visualizar y crear grafos¹⁴. Este software permite una gran cantidad de formatos y tiene una gran variedad de opciones. Dentro de la herramienta desarrollada, Graphviz se ha utilizado para generar grafos en forma de árboles jerárquicos para permitir una visualización más cómoda y compacta de la plantilla que el usuario está generando.

Para implementar esta funcionalidad dentro de la herramienta, el proceso que se ha seguido ha consistido en recorrer de forma recursiva el conjunto de bloques que el usuario ha añadido a su plantilla para, posteriormente, generar un fichero *.dot* que servirá como entrada a Graphviz. En este fichero se describe la estructura que tendrá el grafo que se desea generar con Graphviz, definiendo en primera instancia el tipo y estilo de grafo y, posteriormente, declarando cada uno de los nodos que van a aparecer en el grafo y la relación que existe entre cada uno de ellos.

A continuación se muestra un ejemplo del contenido un fichero *.dot* generado por la herramienta y el grafo generado por Graphviz a partir de dicho fichero:

```
graph G {
    node [shape=oval,fontsize=9,margin=0.01];
    node0 [label="Header"];
    node1 [label="Patient"];
    node2 [label="Structured-body"];
    node3 [label="Component"];
    node4 [label="Section"];
    node5 [label="Component"];
    node6 [label="Component"];

    "node0"-"node1";
    "node1"-"node2";
    "node2"-"node3";
    "node3"-"node4";
    "node2"-"node5";
    "node2"-"node6";
}
```

Imagen 1 – Ejemplo de fichero *.dot*

¹⁴ Grafo: Conjunto de nodos unidos por aristas que representan relaciones binarias.

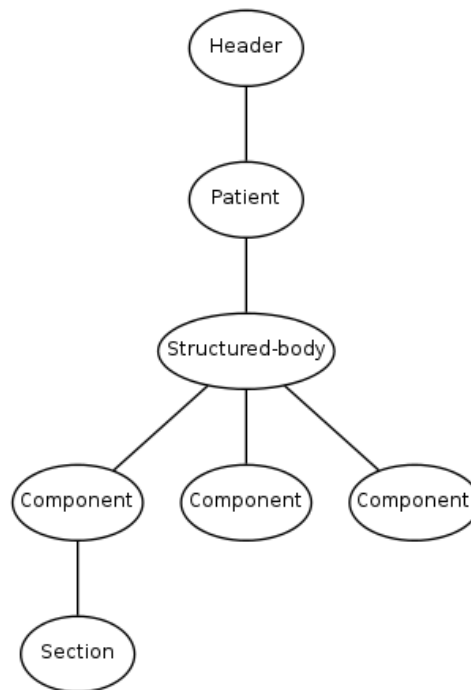


Imagen 2 – Ejemplo de grafo generado por Graphviz

5.5 Librerías de Javascript

En esta sección se explican las librerías de Javascript que usa la herramienta desarrollada para implementar algunas de sus funcionalidades o bien para configurar algún elemento de la interfaz. Todas estas librerías se componen de uno o varios ficheros escritos en Javascript, y algunos de ellos incluyen ficheros `.css` para configurar el estilo. Es necesario indicar que ninguna de ellas ha sido generada o modificada por el alumno.

5.5.1 JQuery

JQuery es una librería que facilita en gran medida la interacción tanto entre los elementos HTML¹⁵ de la página web como entre el cliente y el servidor. Esta librería ha sido utilizada dentro de la herramienta para implementar la comunicación con el servidor, para acceder más fácilmente a algunos elementos HTML de la página y para modificar el estilo de algunos elementos, como es el caso de la caja con sugerencias que

¹⁵ HTML: HyperText Markup Language



se despliega al insertar texto en la caja de texto de la esquina superior derecha de la interfaz gráfica de la herramienta.

5.5.2 Fancybox

Fancybox es otra librería que ha sido creada basándose en JQuery y que se ha utilizado para implementar las partes de la herramienta en las que se necesita simular el comportamiento de una ventana emergente, tal y como sucede en las aplicaciones de escritorio. Estas “ventanas” se han utilizado tanto a la hora de mostrar errores o advertencias como en las ocasiones en las que se muestra el árbol o el texto generado a partir de los bloques introducidos por el usuario. Otra de las características de esta librería que han sido utilizadas dentro de la herramienta es la animación que aparece cuando la herramienta está ocupada realizando alguna operación.

5.5.3 Bootstrap

Bootstrap es una combinación de ficheros Javascript y .css que se han utilizado para aplicar un estilo concreto a la interfaz gráfica de la herramienta. Casi todos los elementos de la interfaz han sido creados utilizando esta librería, incluyendo toda la barra de herramientas superior y el estilo de los bloques que pueden ser arrastrados.

5.5.4 Codemirror

Codemirror es una librería que permite incluir un editor de texto dentro de tu página web. Esta librería cuenta con un gran número de lenguajes y formatos, así como una gran cantidad de opciones para modificar el editor dentro de la página. En este caso, la librería se ha utilizado para el lenguaje XML, que es el lenguaje en el que se generan las plantillas. Además, en la herramienta no se ha implementado un editor de texto, sino que simplemente se ha utilizado la librería a la hora de mostrar código XML en pantalla, poniendo el texto mostrado en diferentes colores y facilitando en gran medida la legibilidad del código.



A continuación se puede ver un fragmento de código perteneciente a una plantilla generada por la herramienta, aplicando diferentes colores al texto usando esta librería:

```
<component>
  <structuredBody>
    <component>
      <section>
        <!-- ALL THIS PART IS OPTIONAL -->
        <templateId root="%%TEMPLATE_ID%%"></templateId>
        <code code="%%CONCEPT_CODE%%"
displayName="%%CONCEPT_LABEL%%" codeSystem="%%VOCABULARY_ID%%"
codeSystemName="%%VOCABULARY_NAME%%"></code>
        <title>SECTION_TITLE</title>
        <text>DESCRIPTIVE_TEXT</text>
        <!-- ***** -->
      </section>
    </component>
  </structuredBody>
</component>
```

Imagen 3 – Ejemplo de texto con formato usando Codemirror

6. LENGUAJES DE PROGRAMACIÓN

En esta sección se indican todos los lenguajes de programación que se han utilizado a la hora de desarrollar la herramienta, así como para qué se ha usado cada uno de ellos. Ha sido necesario el uso de varios lenguajes de programación debido a varias razones:

1. La herramienta se divide en dos componentes (cliente y servidor)
2. Se necesita HTML junto a CSS para crear la interfaz de usuario
3. Algunas de las funcionalidades requieren el uso de bases de datos y, por tanto, el uso de SQL.



6.1 Nodejs

Nodejs es un lenguaje de programación comúnmente utilizado para programar servidores web. Este lenguaje permite crear de una forma sencilla, rápida y eficiente servidores de varios tipos (FTP, HTTP, TCP...) a los que, usando una serie de módulos disponibles desde su propia web, se les pueden añadir una gran cantidad de funcionalidades. Su sintaxis es básicamente la misma que la del lenguaje Javascript, añadiéndole además algunas funciones propias de Nodejs.

Este lenguaje se ha utilizado para implementar la parte del servidor, que se encarga de servir las peticiones procedentes del cliente, que dependerán a su vez de las funciones de la herramienta que vaya utilizando el usuario.

Como se ha indicado anteriormente, se pueden añadir módulos al servidor para añadir funcionalidad extra al servidor. A continuación se muestra una tabla con los módulos que se han usado para implementar el servidor y una descripción de la utilidad que se le ha dado:

Módulo	Utilidad
socket.io & express	Estos dos módulos se han utilizado para crear el servidor en sí, que aceptará peticiones por el puerto 8077
fs	Este módulo facilita una serie de funciones para realizar operaciones con el sistema de ficheros
mysql	Este módulo se ha utilizado cuando ha sido necesario realizar consultas a bases de datos
sys & child_process	Estos dos módulos se han utilizado para poder ejecutar comandos del sistema operativo

Tabla 2 – Módulos de Nodejs

6.2 Otros lenguajes

Para implementar la parte del cliente, es decir, el código que se ejecuta o se muestra en el navegador del usuario han sido necesarios varios lenguajes, ya que es en la parte del cliente donde realmente se programa la interacción entre el usuario y la interfaz y la comunicación con el servidor.



Los lenguajes que se han utilizado en el desarrollo de esta parte se describen a continuación:

- **Javascript:** con este lenguaje se ha implementado toda la interacción con los elementos que componen la interfaz gráfica, es decir, la actualización de la interfaz y la interpretación de las acciones del usuario, además de las peticiones al servidor. Por otro lado, como ya se ha mencionado, se han utilizado también un conjunto de librerías escritas en este mismo lenguaje para añadir alguna funcionalidad al cliente.
- **HyperText Markup Language (HTML):** con este lenguaje se ha diseñado la única página web de la que consta la interfaz gráfica de la herramienta, es decir, el fichero “index.html”. En este fichero se definen todos los elementos que componen la interfaz, que se va actualizando y completando gracias a la interacción con los scripts escritos en Javascript.
- **Cascading Style Sheets (CSS):** este lenguaje es utilizado junto con HTML y Javascript para añadir estilos a los elementos de la interfaz. Principalmente se ha utilizado la librería Bootstrap, aunque también ha sido necesario definir algunas clases en este lenguaje para algunos elementos concretos de la interfaz.
- **Structured Query Language (SQL):** este lenguaje se ha utilizado para realizar las consultas necesarias a la base de datos descrita en la sección 4, es decir, en las opciones en las que el usuario desea saber qué bloques puede utilizar con un concepto clínico dado.

7. ANÁLISIS DEL SISTEMA

7.1 Requisitos

En esta sección se explican los requisitos descritos previamente al desarrollo de la herramienta y que eran necesarios seguir para cumplir los objetivos que se pretendían.

La lista de requisitos se describe a continuación:



1. Interfaz en inglés

Dado que la herramienta desarrollada va a ser utilizada en primera instancia en un proyecto europeo (EURECA), era imprescindible que la interfaz fuera íntegramente desarrollada en inglés para facilitar el uso de la misma a los usuarios, ya que el inglés es uno de los idiomas más hablados.

2. Aplicación web

Otro de los requisitos era que la herramienta fuese una aplicación web, facilitando así el uso de la misma, ya que lo único necesario para poder utilizarla es tener instalado un navegador con HTML5¹⁶, no siendo necesario instalar ningún otro componente en el ordenador del usuario y siendo totalmente independiente del sistema operativo que este use.

3. Uso de bloques atómicos

Este era otro de los requisitos imprescindibles, pues la idea inicial del TFG consistía en ir componiendo una plantilla para un mensaje HL7v3 añadiendo bloques atómicos a una estructura base. Para implementar este requisito ha sido necesario estudiar las partes de un mensaje que pueden ser atómicas (indivisibles) y también las partes del mismo que son opcionales, las cuales son indicadas al usuario a través de un comentario incluido en el código del bloque correspondiente.

Para cumplir con este requisito se han definido los siguientes bloques atómicos:

- Header: En este bloque se define la cabecera del mensaje HL7v3. Dicho bloque forma parte de la estructura base del mensaje, por lo que no puede ser incluido dentro de otro bloque, ni tampoco se le pueden añadir bloques.
- Patient: Este bloque contiene la información acerca del paciente al que corresponde el mensaje HL7v3. Este bloque también forma parte de la base del mensaje, por lo que no puede pertenecer a otro bloque ni se le pueden añadir más bloques.
- Structured body: Este bloque podría ser considerado el bloque raíz, ya que es en él en el que el usuario puede ir insertando bloques y, además, todos los bloques serán hijos de este. Este bloque no contiene información, simplemente se usa para formar correctamente la estructura del mensaje HL7v3.

¹⁶ HTML5: última versión del lenguaje HTML que trae consigo una gran cantidad de mejoras y novedades. Es utilizado ya en la mayoría de los navegadores actuales.



- **Act:** Este bloque representa un acto dentro del mensaje. Puede ser considerado un bloque genérico, ya que los bloques *Observation*, *Procedure* y *SBAM* también son actos, pero además indican qué tipo de acto son.
- **ActRelationship:** Este bloque se utiliza para añadir relaciones entre diferentes actos del mensaje.
- **Component:** Este bloque se utiliza para diferenciar varios componentes dentro del mensaje HL7v3.
- **DoseQuantity:** Este bloque se usa para representar la cuantía de la dosis de algún medicamento que se ha administrado al paciente.
- **Entry:** Este bloque no contiene información, simplemente se usa para poder insertar actos dentro del mensaje.
- **Frequency:** Este bloque se utiliza para representar la frecuencia con que se administra un medicamento o tratamiento.
- **InterpretationCode:** Este bloque se usa para representar una interpretación de un concepto.
- **MethodCode:** Este bloque se usa para representar conceptos que se refieren a métodos.
- **Observation:** Este bloque se usa para incluir en el mensaje actos que son observaciones.
- **Participant:** Este bloque se utiliza para añadir un participante dentro de un acto del mensaje.
- **Period:** Con este bloque se representa la duración de un acto.
- **Procedure:** Este bloque representa actos que son procedimientos.
- **RateQuantity:** Este bloque representa la cantidad de medicamento administrado a un paciente.
- **RouteCode:** Con este bloque se representa la forma de acceder a un órgano de un paciente cuando se le administra un tratamiento.



- SBADM: Este bloque representa actos que son administraciones de sustancias.
- Section: Este bloque no contiene información en sí, se utiliza para dividir un componente en secciones.
- TargetSite: Con este bloque se representa la localización de la dolencia o el lugar en el que se administra un tratamiento a un paciente.
- Translation: Este bloque se utiliza para representar la traducción de un concepto en otro vocabulario clínico distinto del que viene representado en el mensaje.
- Value: Con este bloque se especifica un valor y el tipo del mismo dentro de un acto.

4. Restricciones en el uso de los bloques

Dado que es imprescindible que las plantillas generadas sigan los estándares CDA y HL7v3, se hace necesario poner al usuario una serie de restricciones a la hora de incluir un bloque dentro de otro.

Para cumplir con este requisito se definen las siguientes restricciones atendiendo al tipo de bloque que se está usando:

Bloque	Bloques a los que puede pertenecer	Bloques que puede contener
Header	---	---
Patient	---	---
Structured body	---	Component
Act	Entry ActRelationship	ActRelationship InterpretationCode MethodCode Participant TargetSite Translation
ActRelationship	Act Observation Procedure SBADM	Act Observation Procedure SBADM
Component	Structured body	Section



DoseQuantity	SBADM	---
Entry	Section	Act Observation Procedure SBADM
Frequency	SBADM	---
InterpretationCode	Act Observation Procedure SBADM	---
MethodCode	Act Observation Procedure SBADM	---
Observation	Entry ActRelationship	ActRelationship InterpretationCode MethodCode Participant TargetSite Translation Value
Participant	Act Observation Procedure SBADM	---
Period	SBADM	
Procedure	Entry ActRelationship	ActRelationship InterpretationCode MethodCode Participant TargetSite Translation
RateQuantity	SBADM	---
RouteCode	SBADM	---
SBADM	Entry ActRelationship	ActRelationship InterpretationCode MethodCode Participant TargetSite Translation DoseQuantity Frequency Period RateQuantity RouteCode



Section	Component	Entry
TargetSite	Act	---
	Observation	
	Procedure	
	SBADM	
Translation	Act	---
	Observation	
	Procedure	
	SBADM	
Value	Observation	---

Tabla 3 –Restricciones entre bloques

5. Filtro de bloques por conceptos

Otro de los requisitos necesarios era que la herramienta permitiera indicar al usuario qué bloques puede utilizar dado un concepto.

Para cumplir con este requisito ha sido necesario hacer una consulta a la base de datos descrita en el apartado 4 para determinar las clases a las que pertenece dicho concepto y, una vez determinadas las clases, se indica al usuario los bloques que se pueden utilizar con dichas clases.

Existen dos casos especiales: si el concepto pertenece a la clase *Observation* es necesario realizar una nueva consulta a la base de datos para determinar si el concepto puede tomar valor si se considera como una observación y, en la misma línea, si el concepto pertenece a la clase *SubstanceAdministration* o a la clase *Entity* también será necesario determinar si el concepto puede tomar valor o no. Si la consulta determina que el concepto puede tomar valor será necesario mostrar al usuario el bloque de *Value* (en el caso de ser una observación) o el bloque *DoseQuantity* si se trata del segundo caso.

7.2 Casos de uso

En esta sección se muestran un par de casos de uso en los que se puede ver la secuencia de acciones que se producen dentro de la herramienta cuando se realiza alguna acción concreta.

En los ejemplos expuestos aparecen los siguientes cuatro actores:

1. Usuario: Representa al usuario que está utilizando la herramienta.
2. Cliente: Representa al script que controla la interacción del usuario con la interfaz y realiza peticiones al servidor.



3. Servidor: Representa al servidor utilizado por la herramienta.
4. Base de datos: Representa a la base de datos en la que están almacenados los conceptos clínicos y toda la información relativa a los mismos.

En el primer caso de uso se muestra lo que ocurre cuando el usuario ha tecleado un concepto clínico y pulsa en el botón “*Get info*” para obtener la información relacionada con dicho concepto:

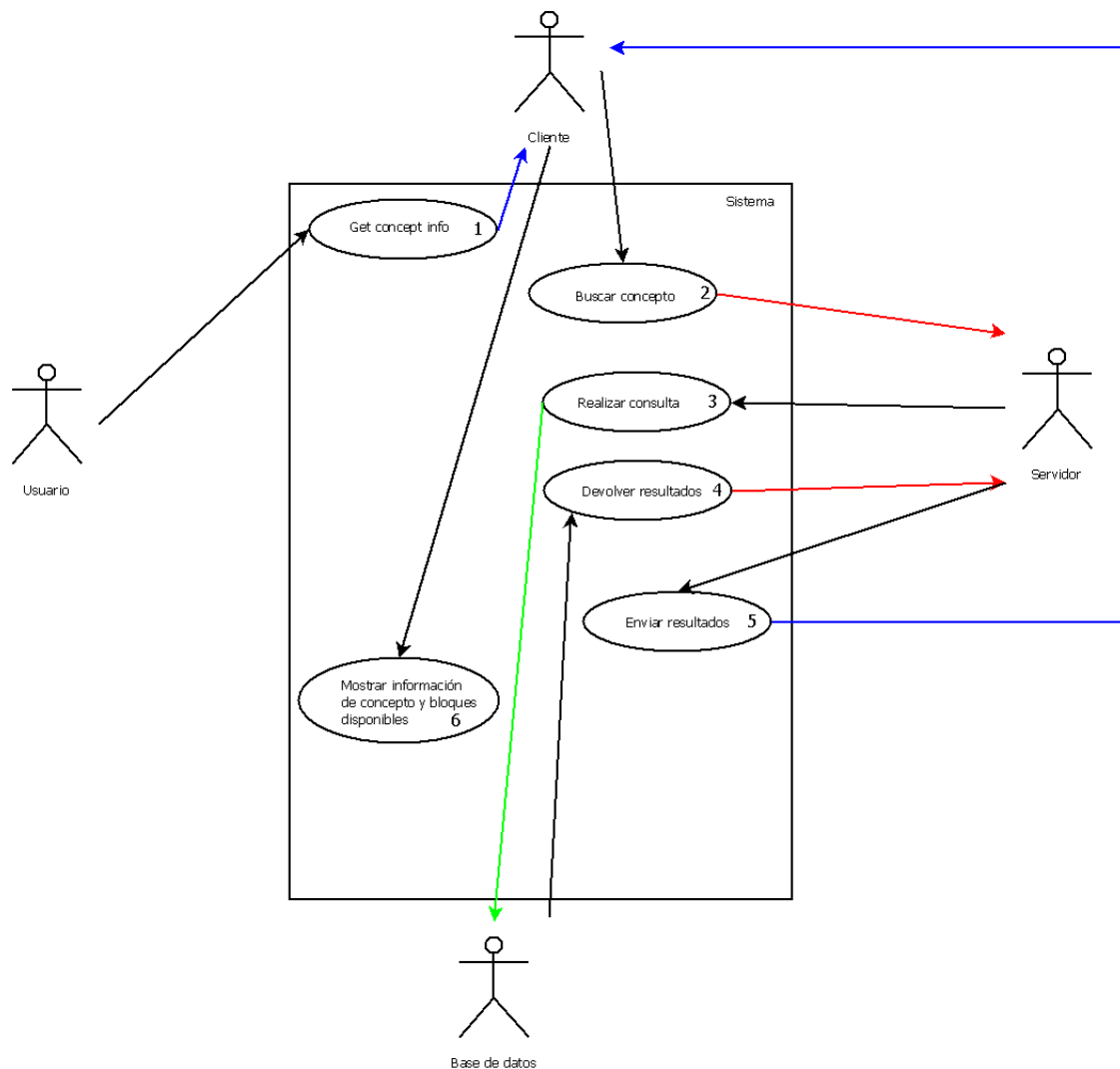


Imagen 4 – Caso de uso 1



En el segundo caso de uso se representan dos acciones distintas realizadas por el usuario. Primero, el usuario arrastra e intenta soltar un bloque. Cuando el bloque ha sido insertado o rechazado por no cumplir las restricciones, el usuario hace click en el botón de “*Show template text*” para ver el código XML de la plantilla completa:

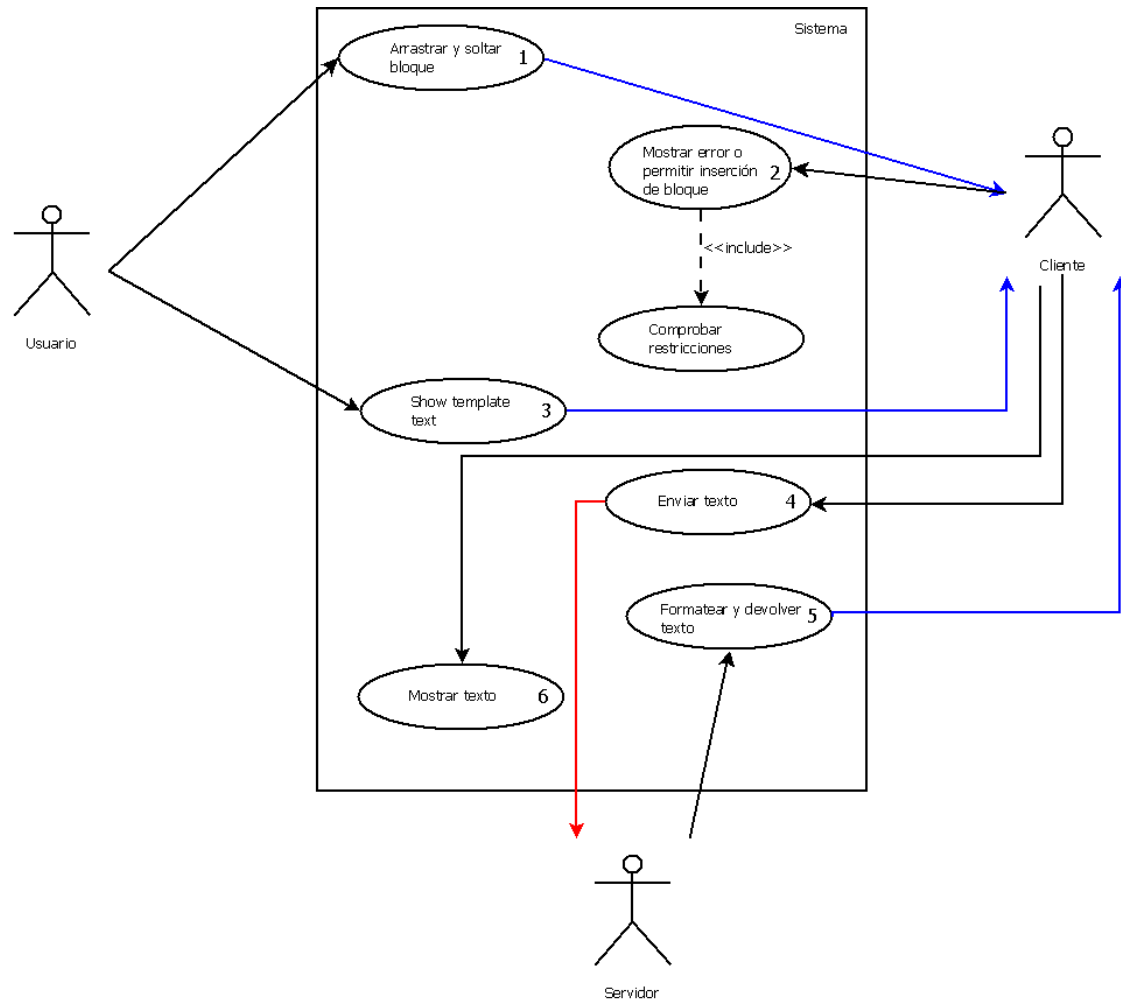


Imagen 5 – Caso de uso 2



8. DISEÑO E IMPLEMENTACIÓN

8.1 Funcionalidad

En esta sección se explican todas las características y opciones que permite realizar la interfaz gráfica de la herramienta para modificar o visualizar la plantilla que está siendo generada por el usuario.

1. **Get info:** Esta opción muestra al usuario información acerca de las clases (y atributos) a las que pertenece un concepto y, además, le indica los bloques que puede utilizar con dicho concepto.
2. **Filter blocks (dado un concepto):** Esta opción permite ocultar los bloques que no pueden ser utilizados con un concepto dado, dejando solamente disponibles aquéllos que sí se pueden utilizar.
3. **Autocompletado de términos:** Esta es una característica que permite mostrar una serie de sugerencias de conceptos clínicos a medida que el usuario va escribiendo texto.
4. **Remove block:** Esta opción permite al usuario eliminar un único bloque que ha sido insertado o en la plantilla. Si dicho bloque contiene otros bloques, se eliminarán también estos.
5. **Filter blocks (dado un bloque):** Esta opción permite al usuario ocultar los bloques que no pueden introducirse dentro del bloque en el que se ha pulsado este botón, dejando a la vista solamente aquéllos que sí se pueden insertar dentro de dicho bloque.
6. **Show block code:** Esta opción, del mismo modo que la opción de *Remove block* permite al usuario mostrar el código XML de un único bloque. Si este bloque contiene otros bloques, se mostrará también el código de éstos.
7. **Restore blocks:** Esta opción permite volver a mostrar todos los bloques una vez que se ha ocultado alguno de ellos tras haber filtrado los bloques, ya sea por concepto o por bloque.



8. Help: Esta opción muestra una ventana emergente con un texto en el que se indica la estructura general de un mensaje HL7v3.
9. Reset: Esta opción permite al usuario restablecer el contenido de la plantilla a su estado inicial, es decir, conteniendo únicamente los bloques de *Header*, *Patient* y *Structured-Body*.
10. Undo: Esta opción permite al usuario deshacer la última modificación realizada a la plantilla. El número de veces que puede utilizarse esta opción es ilimitado.
11. Show template tree: Esta opción muestra una ventana emergente con una imagen de un árbol jerárquico (ver sección 5.4 *Graphviz*) que facilita al usuario la visión de la estructura de la plantilla que está generando.
12. Show template text: Esta opción muestra una ventana emergente con el código XML de la plantilla que ha sido generada hasta ese momento.
13. Save to a file: Esta opción permite descargar un fichero XML con el contenido de la plantilla generada hasta ese momento.

8.2 Estructura interna

Como ya se ha indicado a lo largo del documento, la herramienta está claramente dividida en dos partes: cliente y servidor. En esta sección se indican los ficheros y carpetas que componen la herramienta y la finalidad de cada uno:

1. Parte del servidor:
 - *server.js*: Este fichero codificado con Nodejs es la implementación del servidor en sí.
 - *config.json*: En este fichero se definen algunas características del servidor (puerto, host, contraseña de la base de datos...). Al poner estas características en un fichero separado se facilita el despliegue de la herramienta en distintas máquinas, ya que la configuración de la herramienta consistiría en modificar únicamente este fichero.



2. Parte del cliente:

- Fichero *index.html*: En este fichero, escrito en HTML, se definen todos los elementos de la interfaz y se importan todas las librerías y hojas de estilo que necesita la herramienta.
- Carpeta *blocks*: En esta carpeta están incluidos los ficheros que contiene el código XML de cada uno de los bloques que se utilizan dentro de la herramienta.
- Carpeta *css*: En esta carpeta se incluyen todas las hojas de estilo que usa la herramienta (ficheros *.css*).
- Carpeta *diagrams*: Esta es una carpeta temporal en la que se guardan las imágenes de los árboles jerárquicos mientras están siendo mostradas al usuario.
- Carpeta *dots*: Esta es otra carpeta temporal en la que se guardan los ficheros *.dot* que contienen el código que se le pasa a la herramienta Graphviz para generar los árboles.
- Carpeta *img*: En esta carpeta se guardan todos los iconos utilizados en la interfaz gráfica.
- Carpeta *scripts*: En esta carpeta se guardan todos los ficheros Javascript que implementan la funcionalidad de la herramienta.
- Carpeta *text*: Esta es otra carpeta temporal en la que se guardan los ficheros con el código XML de la plantilla completa o de uno o varios bloques.
- Carpeta *temp*: Esta es otra carpeta temporal. En ella se guardan ficheros con el texto que se manda desde el cliente y que se usará como entrada a xmllint para indentarse.



8.3 Interfaz de usuario

En esta sección se explican todos los elementos que forman la interfaz de la herramienta, además, se añaden una serie de capturas de pantalla para mostrar cada uno de estos elementos. A continuación se enumeran cada uno de estos elementos:

1. Barra de herramientas superior: Esta barra de herramientas se mantiene fija en la parte superior aunque se haga *scroll* en la página. Desde ella se puede acceder a todas las opciones de que dispone la herramienta, excepto la de *Restore blocks*.



Imagen 6 – Barra de herramientas superior

Desde esta barra se accede también a las opciones de *Get info* y *Filter blocks*, que realizan acciones en función del contenido de la caja de texto que se encuentra a su izquierda.

Al hacer click en *Template actions* se despliega un menú en el que se muestran distintas acciones que se pueden aplicar a la plantilla, así como los accesos directos de teclado que corresponden a cada una de ellas.

2. Bloques para definir secciones: Este elemento está compuesto por una “caja” que contiene tres bloques que no se usan para añadir información, sino que sirven para modificar la estructura de la plantilla. Estos bloques pueden ser arrastrados para añadirlos a la plantilla.

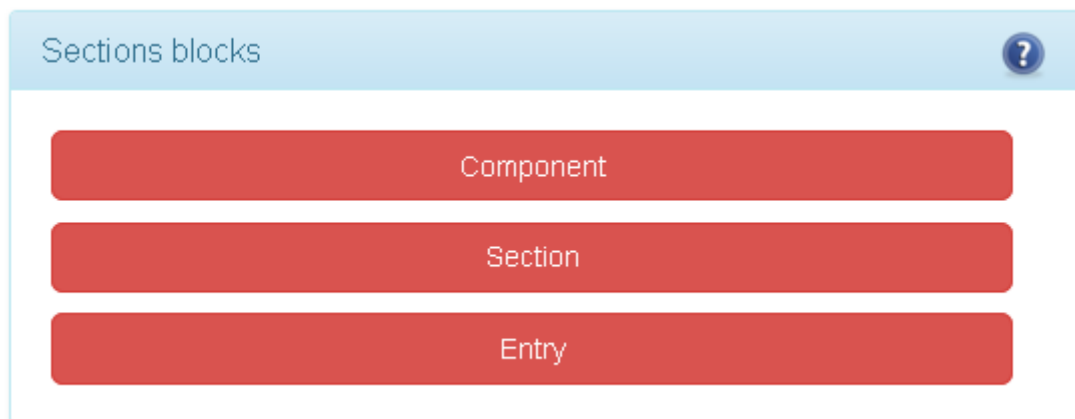


Imagen 7 – Caja con bloques de sección

Este elemento contiene un botón en la parte superior derecha que, como ya se ha mencionado, muestra una ventana emergente en la que se explica la estructura básica de un mensaje HL7v3.



3. Bloques de datos: Este elemento está también formado por una “caja” que contiene todos los bloques que son utilizados para insertar datos. Estos bloques pueden ser arrastrados para añadirlos a la plantilla.

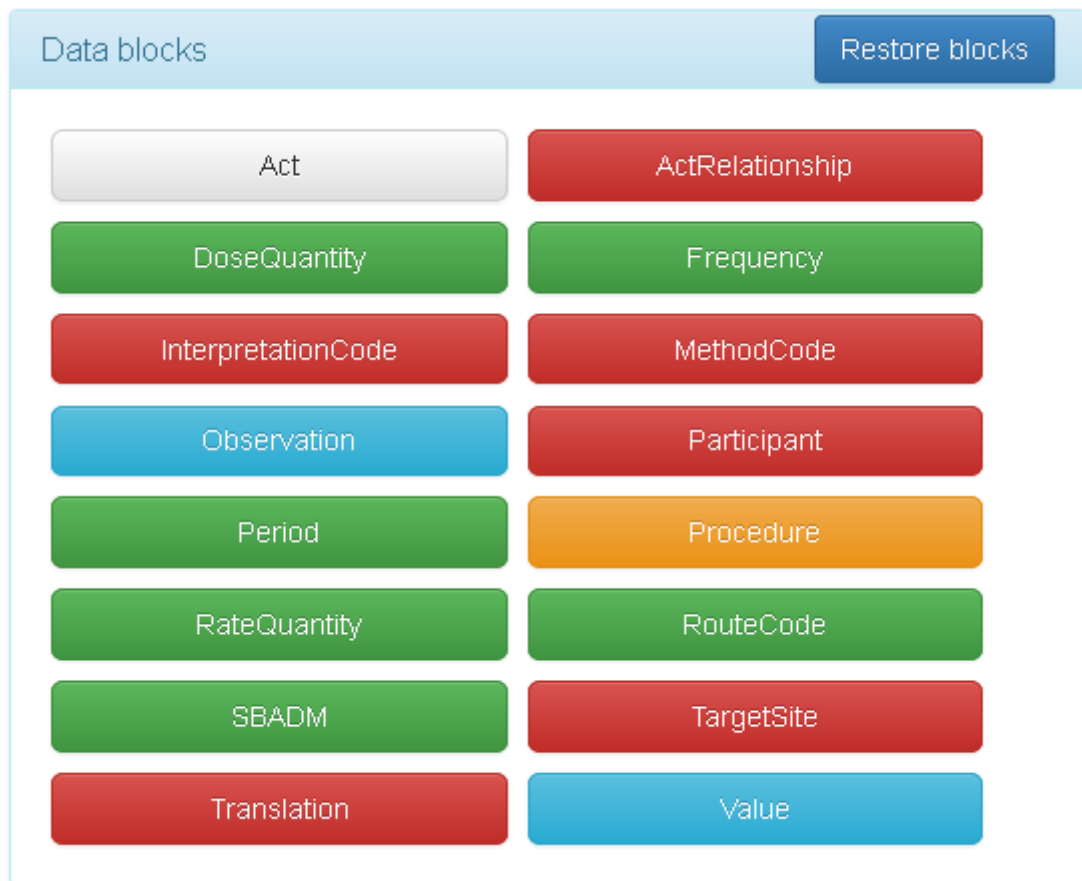


Imagen 8 – Caja con bloques de datos

Como se ve en la imagen, estos bloques siguen un código de colores. Los bloques que tienen el mismo color están relacionados:

- Blanco: Únicamente el bloque *Act*
- Rojo: Bloques “genéricos”. Se pueden insertar dentro de cualquier bloque que represente un acto.
- Verde: Bloques asociados a actos que son administraciones de sustancia.
- Naranja: Únicamente el bloque *Procedure*
- Azul: Bloques asociados a actos que son observaciones.

Por otro lado, como ya se ha indicado, en la parte superior hay un botón (*Restore blocks*) que se usa para volver a mostrar todos los bloques.



4. Parte editable: Esta parte es a la que se arrastran los bloques de los dos puntos anteriores y donde se sueltan para ir formando la plantilla.

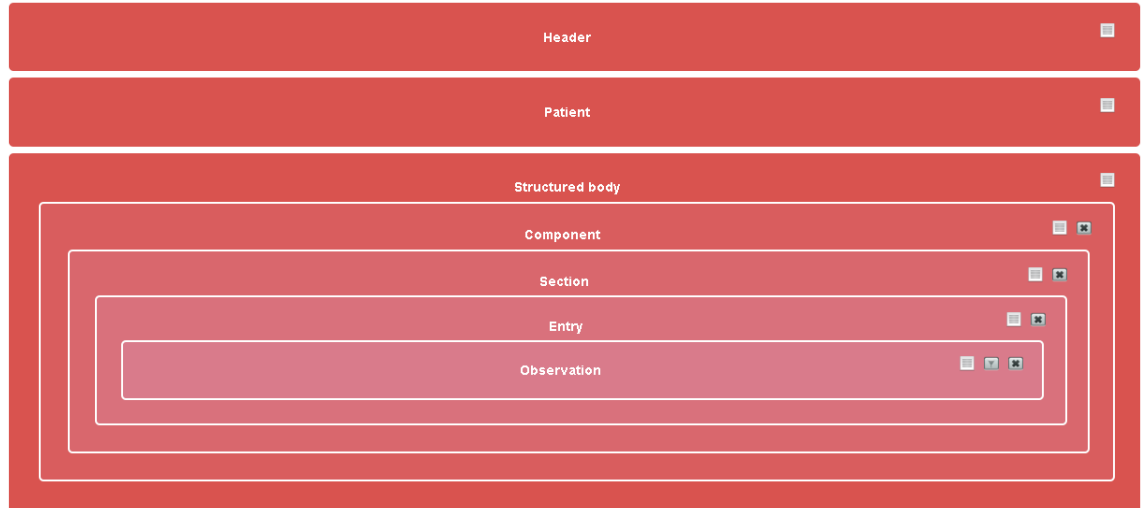


Imagen 9 – Parte editable

Como se ve en la imagen, dentro de cada bloque puede accederse a las acciones que afectan a dicho bloque.

5. Panel lateral de opciones: En este elemento aparecen las mismas opciones que se muestran en el menú desplegable de la barra de herramientas superior. Este menú, al igual que la barra de herramientas superior, permanece fijo aunque se vaya haciendo *scroll* en la página. El motivo de que se hay añadido este panel es que, de esta forma están siempre visibles todas las acciones que se pueden realizar sobre la plantilla.



Imagen 10 – Panel lateral



6. Ventanas emergentes: Este tipo de ventanas se utilizan para mostrar errores y advertencias y para mostrar el código XML de la plantilla o de un bloque, así como los árboles jerárquicos.

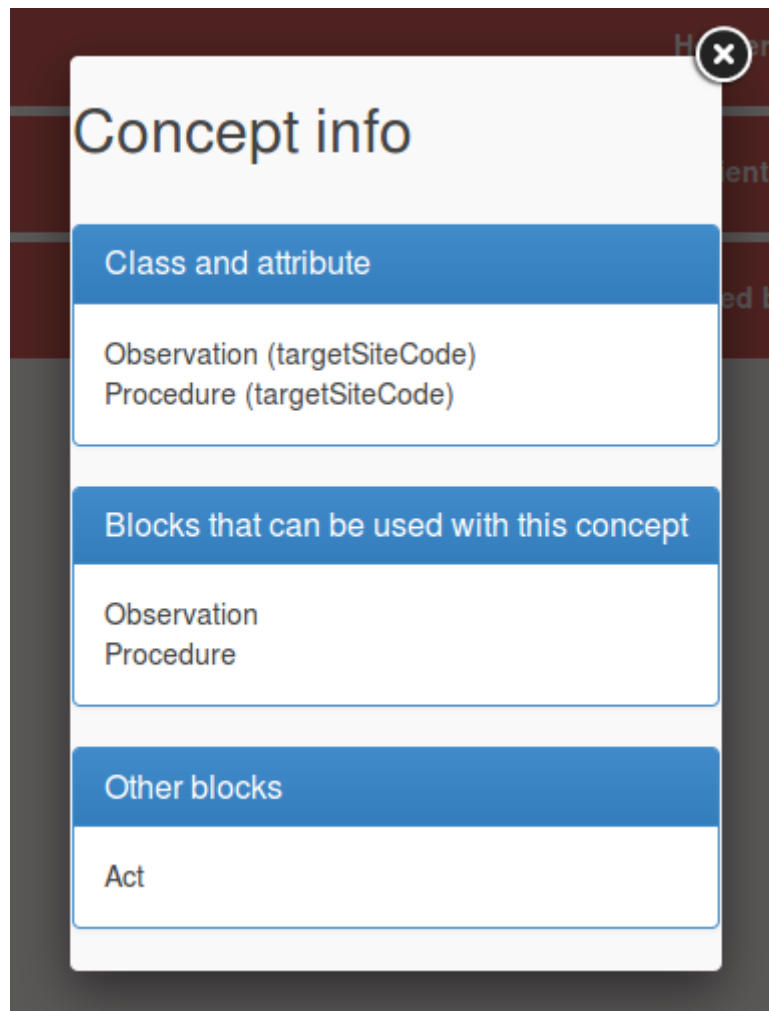


Imagen 11 – Ejemplo de ventana emergente



9. PRUEBAS Y RESULTADOS

Se han realizado diversas pruebas para comprobar el correcto funcionamiento de la herramienta. A continuación se describen el tipo de pruebas que se han realizado y algunos ejemplos concretos de dichas pruebas:

- **Prueba 1:** Validar el código XML de las plantillas generadas en 4 casos distintos. Para realizar dicha validación se ha utilizado un *plugin* perteneciente al programa Notepad++¹⁷. Los diferentes casos se describen a continuación:
 1. Plantilla con una administración de sustancia. Los bloques necesarios para crear esta plantilla son: *Component*, *Section*, *Entry* y *SBADM*, además de los tres bloques que forman la base de la plantilla, es decir, *Header*, *Patien* y *Structured body*.
 2. Plantilla con una observación y un valor para dicha observación. Los bloques necesarios son: *Component*, *Section*, *Entry*, *Observation* y *Value*, además de los tres bloques que forman la base de la plantilla, es decir, *Header*, *Patien* y *Structured body*.
 3. Plantilla con varios bloques *Entry* vacíos. Los bloques necesarios son: *Component*, *Section* y tres bloques *Entry*, además de los tres bloques que forman la base de la plantilla, es decir, *Header*, *Patien* y *Structured body*.
 4. Plantilla con una administración de sustancia relacionada con un acto. Los bloques necesarios son *Component*, *Section*, *Entry*, *Act*, *ActRelationship* y *SBADM*, además de los tres bloques que forman la base de la plantilla, es decir, *Header*, *Patien* y *Structured body*.

Los resultados obtenidos en las 4 pruebas han sido positivos, pues no se han detectado errores al validar ninguno de los 4 ficheros XML.

- **Prueba 2:** Rellenar las 4 plantillas anteriores con datos ficticios y comprobar que los datos se guardan correctamente. Los datos se han incluido de forma manual (ver *Anexo 2 – Ejemplo de mensaje HL7v3 con datos*) y, para realizar la inserción de datos en una base de datos, se ha utilizado el mismo software que se utiliza dentro del proyecto EURECA para cargar datos almacenados en mensajes HL7v3, por lo que podría decirse que las plantillas han sido probadas en un entorno “real”. Para comprobar que las pruebas han sido satisfactorias es necesario verificar que no se ha producido ningún error a la hora de cargar los datos y que toda la información que se había introducido en los mensajes ha sido correctamente almacenada en la base de datos. En los cuatro casos se almacenaron sin errores los datos.

¹⁷ Notepad++: Editor de textos gratuito que soporta una gran variedad de lenguajes.



- Prueba 3: Intentar crear una plantilla errónea, es decir, insertar bloques donde se puede. Durante estas pruebas se ha comprobado que la herramienta no deja insertar un bloque donde no puede ser insertado y que, además, informa correctamente sobre los bloques en los que sí puede ser insertado dicho bloque.
- Prueba 4: Comprobar que los árboles jerárquicos generados por la herramienta son correctos en los cuatro casos descritos en la prueba 1. La comprobación se ha realizado de forma visual, es decir, sin utilizar ningún software que realice pruebas automáticas.
En los cuatro casos el árbol generado con la herramienta se correspondía con los bloques insertados.
- Prueba 5: Del mismo modo que en la prueba anterior, se ha comprobado que el código XML mostrado por la herramienta se corresponde con los bloques insertados. Estas pruebas se han realizado utilizando también los casos descritos en la prueba 1.
- Prueba 6: En esta prueba se ha comprobado que tanto la información mostrada al pulsar en “Get info” como los bloques que se han ocultado al pulsar en “Filter blocks” son correctos para los siguientes conceptos clínicos:
 1. Breast
 2. Carcinoma
 3. Epirubicin
 4. Administration
 5. Size of tumor

Con estos conceptos se ha pretendido cubrir todos los casos posibles, es decir, conceptos que pertenezcan a todas las clases y conceptos que puedan y no puedan tomar valor.

Para comprobar que la herramienta realiza las dos operaciones correctamente se ha comprobado manualmente la base de datos que contiene esta información.

Para los 5 conceptos nombrados anteriormente la información mostrada por la herramienta era acorde con la información almacenada en dicha base de datos.

- Prueba 7: Esta última prueba ha consistido en combinar las acciones de *Undo*, *Reset* y *Remove block* y comprobar que los bloques de la plantilla se iban comportando como se esperaba que lo hicieran.



10. CONCLUSIONES

En esta sección final se describen las conclusiones obtenidas durante el desarrollo del proyecto y se habla de la consecución de los objetivos planteados al inicio del desarrollo del mismo.

La realización de este TFG ha permitido la creación de una herramienta que puede ser de gran utilidad en los proyectos en que se necesite usar mensajes HL7v3 para compartir información. Dado que el uso de las nuevas tecnologías dentro de la medicina va en aumento y que el intercambio de información entre entidades clínicas se va haciendo cada vez más importante para tener una base mayor a la hora de diagnosticar una enfermedad o sugerir un tratamiento, la optimización en el proceso consistente en el intercambio de dichos datos se hace imprescindible. Es en este proceso en el que el uso de herramientas como la que se ha desarrollado durante este proyecto cobra gran importancia, ya que no sólo aumentan la rapidez con la que se generan los mensajes, sino que, además, abstrae en gran medida al usuario de la complejidad de los mismos. Es por estos motivos por los que creo que esta herramienta puede cobrar gran importancia dentro de proyectos como EURECA, aunque habrá que esperar que se extienda su uso para comprobar que la utilidad de la misma es tan alta como se espera.

Por otro lado, en cuanto a la consecución de los objetivos, es necesario decir que se han conseguido todos los objetivos que se pretendían con este TFG y, aunque la herramienta desarrollada tiene un amplio margen de mejora, creo que puede servir como una buena base para el desarrollo de futuras herramientas basadas en ésta o la mejora de ella a partir de lo que ya se ha desarrollado.



BIBLIOGRAFÍA

- [1] Recurso editable por cualquier usuario registrado. *Wiki del proyecto EURECA* [Online]. Disponible en: http://atlas.ics.forth.gr/EURECA/wiki/index.php/Main_Page
****Es necesario ser usuario registrado para acceder al contenido.**
- [2] EURECA Project. (2012). *Web oficial del proyecto EURECA* [Online]. Disponible en: <http://eurecaproject.eu/>
- [3] Recurso editable por cualquier usuario registrado. *Wiki sobre informática biomédica* [Online]. Disponible en: http://informaticabiomedica.wikia.com/wiki/Main_Page
- [4] The ART-DECOR expert group. (2010). *Web oficial del proyecto ART-DECOR* [Online]. Disponible en: <http://art-decor.org/>
- [5] Health Level Seven International. (2007). *Web oficial de HL7* [Online]. Disponible en: <http://www.hl7.org/>
- [6] *HL7 Implementation guide for CDA ® Release 2*, Draft Standard for Trial Use, Julio de 2012.
- [7] Recurso editable por cualquier usuario registrado. *Wikipedia* [Online]. Disponible en: <http://es.wikipedia.org/wiki/Wikipedia:Portada>
- [8] Daniel Veillard. *Web con información sobre xmllint* [Online]. Disponible en: <http://xmlsoft.org/xmllint.html>
- [9] *Web oficial de Graphviz* [Online]. Disponible en: <http://www.graphviz.org/>
- [10] jQuery Foundation. (2014). *Web oficial de jQuery* [Online]. Disponible en: <http://jquery.com/>
- [11] Mark Otto and Jacob Thornton. (2011). *Web oficial de Bootstrap* [Online]. Disponible en: <http://getbootstrap.com/>
- [12] Marijn Haverbeke. *Web oficial de Codemirror* [Online]. Disponible en: <http://codemirror.net/>
- [13] Fancybox Group. (2008). *Web oficial de Fancybox* [Online]. Disponible en: <http://fancybox.net/>
- [14] Joyent, Inc. (2011). *Web oficial de Nodejs* [Online]. Disponible en: <http://nodejs.org/>



[15] Don Ho. (2011). *Web oficial de Notepad++* [Online]. Disponible en:
<http://notepad-plus-plus.org/>

[16] Quality Systems, Inc. (2011). *Web oficial de Mirth Connect* [Online]. Disponible
en: <http://www.mirthcorp.com/products/mirth-connect>



ANEXOS

Anexo 1 – Ejemplo básico de plantilla

```
<?xml version="1.0" encoding="ISO-8859-1" standalone="no"?>
<ClinicalDocument xmlns="urn:hl7-org:v3"
xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
xsi:schemaLocation="urn:hl7-org:v3 CDA.xsd">
  <typeId extension="POCD_HD000040"
root="2.16.840.1.113883.1.3"></typeId>
  <templateId root="%%TEMPLATE_ID%%"></templateId>
  <id extension="1.0"
root="%%type=TYPE;name=TRIAL_NAME;messageId=MESSAGE_ID%%"></id>
  <title>%%TITLE_FOR_THIS_DOCUMENT%%</title>
  <effectiveTime
value="%%DATE_OF_THE_CREATION_OF_THIS_DOCUMENT(YYYYMMDDHHMMSS)%%"></ef
fectiveTime>
  <confidentialityCode code="R"
codeSystem="2.16.840.1.113883.5.25"></confidentialityCode>
  <languageCode code="en-US"></languageCode>
  <setId extension="1.0"
root="2.25.299518904337880959076241620201932965147.6.1"></setId>
  <versionNumber value="1"></versionNumber>
  <recordTarget>
    <patientRole>
      <id extension="%%PATIENT_ID%%"></id>
      <patient>
        <name>%%PATIENT_NAME%%</name>
        <administrativeGenderCode code="%%M(Male), F(Female), UN
(Un differentiated)%%"
codeSystem="2.16.840.1.113883.5.1"></administrativeGenderCode>
        <birthTime value="%%BIRTHDATE(YYYYMMDDHHMMSS)%%"></birthTime>
      </patient>
    </patientRole>
  </recordTarget>
  <component>
    <structuredBody>
      <component>
        <section>
          <!-- ALL THIS PART IS OPTIONAL -->
          <templateId root="%%TEMPLATE_ID%%"></templateId>
          <code code="%%CONCEPT_CODE%%"
displayName="%%CONCEPT_LABEL%%" codeSystem="%%VOCABULARY_ID%%"
codeSystemName="%%VOCABULARY_NAME%%"></code>
          <title>SECTION_TITLE</title>
          <text>DESCRIPTIVE_TEXT</text>
          <!-- ***** -->
          <entry>
            <observation classCode="OBS" moodCode="EVN">
              <templateId root="%%TEMPLATE_ID%%"></templateId>
              <id root="%%OBSERVATION_ID%%"></id>
```




```

        <code code="%%OBSERVATION_CODE%%"
displayName="%%OBSERVATION_LABEL%%" codeSystem="%%VOCABULARY_ID%%"
codeSystemName="%%VOCABULARY_NAME%%"></code>
        <!-- THIS PART IS OPTIONAL -->
        <text>DESCRIPTIVE_TEXT</text>
        <!-- THIS PART IS OPTIONAL -->
        <statusCode code="%%OBSERVATION_STATUS%%"></statusCode>
        <effectiveTime
value="%%OBSERVATION_DATE(YYYYMMDDHHMMSS) %%"></effectiveTime>
    </observation>
</entry>
</section>
</component>
</structuredBody>
</component>
</ClinicalDocument>

```

Anexo 2 – Ejemplo de mensaje HL7v3 con datos

```

<?xml version="1.0" encoding="ISO-8859-1" standalone="no"?>
<ClinicalDocument xmlns="urn:hl7-org:v3"
xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
xsi:schemaLocation="urn:hl7-org:v3 CDA.xsd">
    <typeId extension="POCD_HD000040" root="2.16.840.1.113883.1.3"/>
    <templateId root="id"/>
    <id extension="1.0"
root="%%type=Prueba;name=Prueba;messageId=1234567abc%%"/>
    <title>PRUEBA DE CARGA DE DATOS</title>
    <effectiveTime value="20140530"/>
    <confidentialityCode code="R" codeSystem="2.16.840.1.113883.5.25"/>
    <languageCode code="en-US"/>
    <setId extension="1.0"
root="2.25.299518904337880959076241620201932965147.6.1"/>
    <versionNumber value="1"/>
    <recordTarget>
        <patientRole>
            <id extension="567890"/>
            <patient>
                <name>Israel Suárez</name>
                <administrativeGenderCode code="M"
codeSystem="2.16.840.1.113883.5.1"/>
                <birthTime value="20140530"/>
            </patient>
        </patientRole>
    </recordTarget>
    <component>
        <structuredBody>
            <component>
                <section>
                    <!-- ALL THIS PART IS OPTIONAL -->
                    <entry>

```



```
<substanceAdministration classCode="SBADM" moodCode="EVN">
  <templateId root="%%TEMPLATE_ID%% (OPTIONAL) "/>
  <id root="%%SBADM_ID%%"/>
  <statusCode code="Complete"/>
  <effectiveTime value="20140530"/>
  <consumable>
    <manufacturedProduct classCode="Class">
      <templateId root="1234567890"/>
      <id root="1234567890"/>
      <manufacturedMaterial>
        <code code="1111111111" displayName="Epirubicin"
codeSystem="2.16.840.1.113883.5.96" codeSystemName="SNOMED"/>
      </manufacturedMaterial>
    </manufacturedProduct>
  </consumable>
</substanceAdministration>
</entry>
</section>
</component>
</structuredBody>
</component>
</ClinicalDocument>
```

Este documento esta firmado por



Firmante	CN=tfgm.fi.upm.es, OU=CCFI, O=Facultad de Informatica - UPM, C=ES
Fecha/Hora	Thu Jun 05 13:43:46 CEST 2014
Emisor del Certificado	EMAILADDRESS=camanager@fi.upm.es, CN=CA Facultad de Informatica, O=Facultad de Informatica - UPM, C=ES
Numero de Serie	630
Metodo	urn:adobe.com:Adobe.PPKLite:adbe.pkcs7.sha1 (Adobe Signature)